DUAL-LAYER VISUAL VOCABULARY TREE HYPOTHESES FOR OBJECT RECOGNITION

Sandra Ober, Martin Winter, Clemens Arth, Horst Bischof

Institute for Computer Graphics and Vision, Graz University of Technology
{ober,winter,arth,bischof}@icg.tu-graz.ac.at

ABSTRACT

This paper introduces an efficient method to substantially increase the recognition performance of a vocabulary tree based recognition system. We propose to enhance the hypothesis obtained by a standard inverse object voting algorithm with reliable descriptor co-occurrences. The algorithm operates on different layers of a standard *k*-means tree benefiting from the advantages of different levels of information abstraction. The visual vocabulary tree shows good results when a large number of distinctive descriptors form a large visual vocabulary. Co-occurrences perform well even on a coarse object representation with a small number of visual words. An arbitration strategy with minimal computational effort combines the specific strengths of the particular representations. We demonstrate the achieved performance boost and robustness to occlusions in a challenging object recognition task.

Index Terms— Machine vision, Object recognition, Image databases, Tree data structures, Clustering methods

1. INTRODUCTION

Recent research interest has focused on the problem of local feature based object recognition in large databases (*e.g.* [1, 2]). Usually the approaches follow the common scheme of interest point detection, descriptor calculation, and matching based on comparison of query object descriptors against the learned training set. One challenge is to organize this huge number of high dimensional descriptors in such a way, that an efficient query is possible. A tree is a well suited data structure used for fast indexing and encouraging recognition rates have been achieved recently (*e.g.* [3, 4]). One typical example is the approach of David Lowe [1], who organized SIFT descriptors from all training images in a kd-tree with a best-bin-first modification to find approximate nearest neighbors to the descriptors of the query. The correspondences of the matched descriptor pairs of the query and kd-tree patches have to be confirmed or rejected in further verification and consistency checks. Obdrzalek and Matas [4] used a binary decision tree to index keypoints and minimize the average time to decision. The leaves of the tree represent a few local image areas where every inner node is associated with a *weak classifier*. Nistér and Stewénius [3] presented an approach where hundred thousands of local descriptors are quantized in a hierarchical vocabulary tree. It is able to organize a database up to 1 million images. They presented a scoring scheme which results have to be verified in an additional post-verification step using the geometry of the matched keypoints of the n top ranked objects to improve the retrieval quality. Common to most of the approaches is the need for such a post-verification algorithm to guarantee for an acceptable performance rate and stable recognition results. A popular method is using geometrical constraints for eliminating false positives and strengthening correct hypotheses [3]. Another possibility are consistency checks of local neighbor relations of interest points [2].

While all of these algorithms require additional computational overhead, the information we incorporate into our system can be obtained almost for free from our own tree-based representation. In particular, we build a hierarchical vocabulary tree and apply inverse voting similar to Nistér and Stewénius [3]. The inverse voting uses the leaves of the tree. Another very coarse representation is taken from a lower tree level. To obtain a high distinctiveness of that coarse representation, we use a very efficient, yet memory and computationally efficient specificity of spatial relations, namely co-occurrences of descriptors. Spatial relations among keypoints have already been investigated by many authors (e.g. [5, 6, 7, 8]) and it has been shown, that they can significantly improve recognition performance. In contrast to other approaches we use a very extreme form of co-occurrences as we represent only the presence or absence of co-occurrences. The intuition behind is that the co-occurrence of descriptors is very discriminative because it is very unlikely, that two neighboring descriptors co-occur just by chance. To foster the hypotheses of inverse voting by co-occurrences we use a rather simple but effective arbitration strategy.

2. DUAL-LAYER TREE HYPOTHESES

2.1. Building the visual vocabulary tree

We use a hierarchical *k*-means tree as data structure for fast indexing and retrieval of descriptors as illustrated in Figure 1. Instead of building the tree with hundred thousands of de-

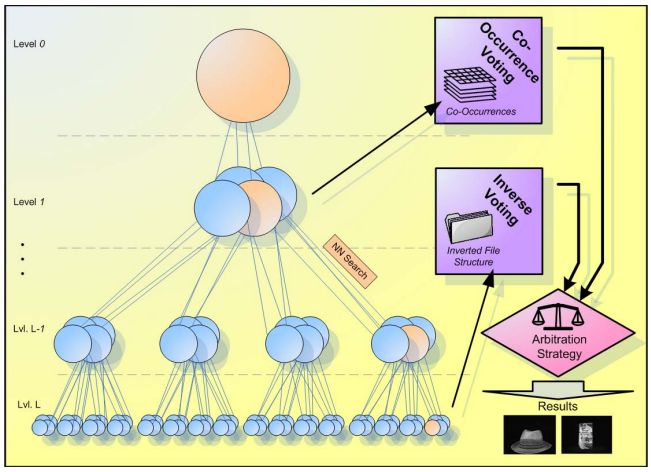

Fig. 1. Depiction of the vocabulary tree (k=4) and tree levels used for inverse voting and co-occurrence representations.

scriptors, we propose a tree, where a lower number of visual words act as leaves, because Nistér and Stewénius have shown in [3], that for more than 100K leaf nodes no substantial performance increase can be expected. So, we quantize the descriptors with unsupervised *agglomerative clustering* using the proposed *Average-Link algorithm with RNNs* of Leibe *et.al.* described in [9]. It has feasible runtime properties and can deal which such a large number of descriptors. The obtained visual words are partitioned in k nodes using k-means and propagated to the next level until no further splitting is possible. Thus we reduce the time spent for building the tree from several days to a few hours.

2.2. Indexing & co-occurrence matrix

For indexing an object, we compute descriptors and choose for each of them the nearest of the k cluster nodes in a deeper level, starting at the root. Every leaf has a unique index and we can represent every image as a set of indices. These are used to store all pre-matches in an *inverted file structure* (IFS). Every index of the IFS is assigned to all object- or image ID-numbers where their descriptors have matched with this leaf. For the verification step, we take the best matching cluster center in a certain already calculated(!) lower k-means tree level (typically $l_c = 2, 3, 4$). Building the vocabulary tree in an off-line calculation, we obtain $n_c = k^{l_c}$ cluster indices in a layer l_c of the tree and branch factor k (number of children for every node). For every keypoint in the image a corresponding cluster index is stored. Note, that there is nearly no additional computational effort necessary to extract that representation out of the tree. To calculate the co-occurrence matrix (with dimensionality n_c), we simply identify the nearest neighbor for every keypoint in image space. Thus, every co-occurrence is identified by a pair of cluster indices which we insert into the two-dimensional co-occurrence matrix. The nearest neighbor property of certain interest points in the image space is sometimes violated by spurious highlights, unstable detection of keypoints and of course aspect changes introduced by different viewpoints. We alleviate this problem by entering the nn nearest neighbors (typically $nn = 3$) in the co-occurrence matrix. Only $1 - 2‰$ of the possible co-occurrences are assigned and multiple occurrences are even much more unlikely. Thus it is possible to limit the entries to the binary information whether a specific co-occurrence is observed for a certain object or not (sparse storage scheme). To build the full representation for a single object (multiple viewpoints) all the co-occurrences of the trained images are entered in one single two-dimensional matrix. Therefore, we have exactly one co-occurrence matrix per object trained.

2.3. Vocabulary tree hypotheses & co-occurrence voting

To recognize an object or image with the vocabulary tree we use the same routine as for indexing. We use the gathered indices with our IFS to set up a scoring table for each object or image. So, the table gives us an object voting list ranked by the number of matched descriptors. The hypothesis has to be normalized by the number of descriptors for each object or image used in the indexing step to achieve fairness for every database object or image to be recognized if the number of descriptors is very low and therefore the occurrences in the IFS are very sparse. Instead of using a separated computationally demanding method to improve retrieval quality, we use a more efficient way to verify the generated hypothesis by co-occurrences gathered online at a lower tree level. Similar to the training step we build the co-occurrence matrix for the query image directly from the cluster indices already associated in the k-means tree on a lower level, and apply deliberately a very simple matching procedure. The matching score is calculated by a simply AND operation of the sparse co-occurrence matrices and by counting the number of resulting matches. So in fact, the matching is only a primitive maximum voting of congruent co-occurrences in the binary matrices.

2.4. Arbitration strategy

The arbitration-component improves the result of the standard inverse voting approach with the additional information obtained by the co-occurrences. As we want to avoid any time consuming adaptation to a specific data set, we apply a heuristic algorithm providing good results. The results of 'inverse voting approach' and 'co-occurrences' cannot be directly combined due to the different matching strategy. So we 'unify' the output to an abstract layer. Each algorithm selects the top ranked object as distinct answer and provides a 'level of significance': 'unambiguous', 'low confidence' and 'unknown' object (see Equation 1). The significance value k of the voting histogram obtained by the inverse object voting

algorithm is easily evaluated by computing the ratio between the number of votes of the first and second ranked objects. Best results are achieved if the thresholds are set to $t_1 = 2$ and $t_2 = 1.6$. The level of significance for co-occurrences is determined by two facts. The first one is an absolute threshold which assigns all objects with less than t co-occurrences to the 'unknown' confidence level (typically $t = 5$). The second one is related to the 'peakedness' of the voting histogram for co-occurrences. As a quantitative estimate for that, we calculate the kurtosis of the discrete voting histogram function. As the 'ideal kurtosis' (impulse function) for a perfect voting histogram is proportional to the number of objects (discrete samples in histogram), we normalize the kurtosis. We can choose the decision thresholds for assignment to different confidence levels (t_1, t_2) relative to the 'ideal kurtosis'. In our experiments we obtained best results by setting $t_1 = 0.33$ and $t_2 = 0.1$. A special handling is required if both approaches have the same level of significance but vote for different objects. In this case, we search for the ranking of each selected object in the other algorithms ranking cue.

$$\left\{ \begin{array}{l} k >= t_1 \Rightarrow `unambiguous' \\ t_2 <= k < t_1 \Rightarrow `low\ confidence' \\ k < t_2 \Rightarrow `unknown' \end{array} \right\} \quad (1)$$

3. EXPERIMENTS

For object recognition we took a subset of 400 objects from the publicly available Amsterdam Library of Object Images (ALOI) [10] and detect Lowe's *Difference of Gaussian* (DoG) detector together with SIFT-keys [1]. The objects were selected with respect to a sufficient number of keypoints detected on the objects surface and the scales of the obtained keypoints were restricted in order to be robust against pixel noise and to avoid the detection of too large regions. We used descriptors from a subset of 100 objects (500 images, $\pm 60^o$ in steps of 30^o) and performed agglomerative clustering. After that, we generated a *k*-means tree with a branch factor of $k = 9$ with about 140.000 visual words.

In order to capture enough variances in the appearances of an object for training, we presented 5 views of 400 objects to the system (2000 images, $\pm 60^o$ in steps of 30^o), where 300 objects presented totally new descriptors. To evaluate the recognition rate we took 13 views from all 400 objects (5200 images, $\pm 60^o$ in steps of 10^o).

3.1. Performance comparisons

In this experiment we investigated the influence of additional information (co-occurrences) obtained from different levels of the tree. In Figure 3(a) the results of our method and the pure inverse voting result on the whole rotation range of 120 degrees are shown. The blue curve describes the results from the standard inverse voting, while the other curves are obtained incorporating information from three different vocabulary tree levels for co-occurrences into our final arbitration strategy. Using additional information from level 2 results in an average performance increase of even 4%. While accessing information from level 3 leads to an overall increase of up to 12% there is no further performance improvement when taking into account information from higher levels (level 4 results in 11%).

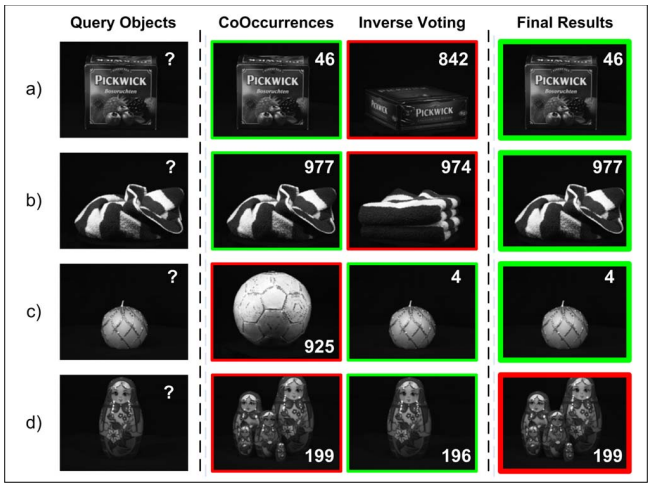

Fig. 2. Four sample query images, the intermediate outcomes and the final results are shown from left to right. While correct hypotheses are marked with a green border, a red border indicates a wrong outcome.

In Figure 2 four query examples, the intermediate results and the final object hypotheses are depicted. In the first two cases, the use of co-occurrences enables our arbitration strategy to draw the right decision even if the inverse voting prefers the wrong object. In the third case the arbitration-module still correctly favors a strong inverse voting result over a weak co-occurrence voting. The last object is labeled incorrectly, but note that the query object is a member of the group of objects chosen.

3.2. Results obtained by occlusion and runtime

To support the claim on the robustness of the obtained performance increase, we made some experiments with varying partially occluded objects. We simulate the occlusions by removing a substantial part of the objects appearance applying a black rectangle. As the objects of the ALOI database are not normalized with respect to their appearance size, we determine the relative area of occlusion for each object separately with respect to the lateral cut of the particular object observed. Figure 3(b) shows the mean recall rates for a different amount of occlusions. The mean recall rates for the (standard) inverse voting approach and our combined approach (arbitration strategy) remain rather stable up to an oc-

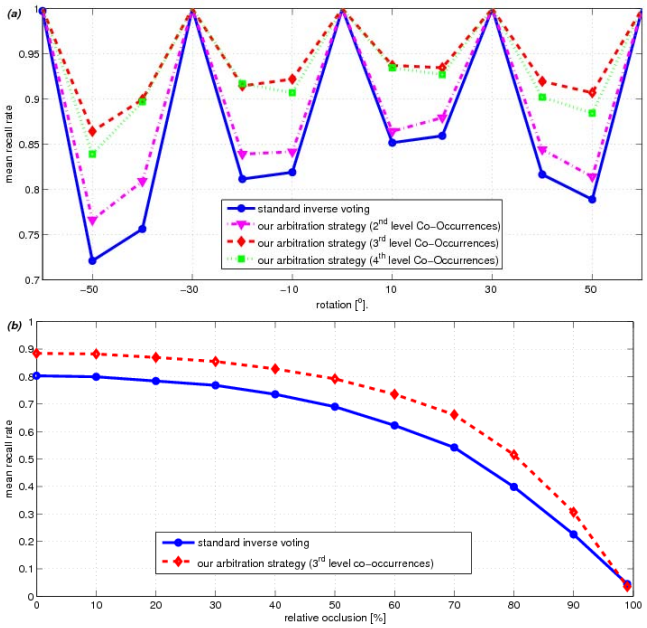

Fig. 3. Comparison of different recall rates for different tree levels used for the co-occurrence matrix *(a)* and Mean recall rates of the inverse voting approach and our arbitration strategy for a different amount of occlusions *(b)* .

clusion about 40%. The performance increase by our combined approach is also constant about 8-10% with respect to the standard approach for all tested occlusions. It is possible to correctly identify the objects even with a fistful number of descriptors. A motivation for the usage of weaker descriptors comes from the fact, that the recognition speed of the current implementation is limited by the runtime of the already highly optimized C/C++ implementations of keypoint detection (DoG) and their descriptors (SIFT) [1] while our approach is currently implemented in MATLAB. Table 1 gives a raw over-view about the relative runtime effort spent in different components of the recognition stage. Only 17.4% of the overall burden for recognition is used for the assignment of obtained query descriptors to the corresponding cluster indices. The computational costs for the voting and arbitration strategy parts are nearly negligible. Reducing the calculation effort for keypoint detection and descriptor calculation by coevally obtaining high recognition performance would further improve the efficiency of our approach.

4. CONCLUSION AND FUTURE WORK

In this paper we have introduced a new method to increase the recognition performance of a vocabulary tree based recognition system. We improved the hypotheses of an inverse object voting algorithm by a very simple specificity of spatial relations, namely descriptor co-occurrences. A rather heuristic

component	runtime (ms)	%
DoG & SIFT calculation	3351	80.6
assignment (tree propagation)	723	17.4
inverse voting	62	1.5
co-occurrences	15	0.4
arbitration strategy	4	0.1

Table 1. Mean runtimes of certain recognition components obtained on a Intel Xeon 2.80GHz CPU.

but powerful arbitration strategy with minimal computational effort combines the specific strengths of the particular representations. The achieved increase of performance has been demonstrated in a challenging object recognition task and we have also shown the robustness of the approach even for a substantial amount of occlusions. The main advantage of our approach is the fact, that we use two different levels of information abstraction provided in various layers of the tree. Thus we can avoid the calculation of an additional representation for the descriptor co-occurrences. As the main computational burden of the recognition system is carried by calculation of the keypoints and their descriptors, in future research we will use our approach to work with even weaker detectors and descriptors but keeping recall rates high by combination of two or more levels of information abstraction.

5. REFERENCES

[1] David Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, pp. 91–110, 2004.

[2] Josef Sivic and Andrew Zisserman, "Video google: a text retrieval approach to object matching in videos," in *ICCV*, 2003, pp. 1470–1477 vol.2.

[3] David Nistér and Henrik Stewénius, "Scalable recognition with a vocabulary tree.," in *CVPR*, 2006, pp. 2161–2168.

[4] Stepan Obdrzalek and Jiri Matas, "Sub-linear indexing for large scale object recognition," in *BMVC*, 2005, vol. 2.

[5] Gustavo Carneiro and David Lowe, "Sparse flexible models of local features," in *ECCV*, 2006, vol. 3953, pp. 29–43.

[6] David J. Crandall and Daniel P. Huttenlocher, "Weakly supervised learning of part-based spatial models for visual object recognition," in *ECCV*. 2006, vol. 3951, pp. 16–29, Springer.

[7] Rob Fergus, Pietro Perona, and Andrew Zisserman, "A sparse object category model for efficient learning and exhaustive recognition," in *CVPR*, 2005.

[8] Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman, and William T. Freeman, "Discovering objects and their location in images," in *ICCV*, 2005, vol. 1, pp. 370–377.

[9] Bastian Leibe, Krystian Mikolajczyk, and Bernt Schiele, "Efficient clustering and matching for object class re-cognition," in *BMVC*, 2006.

[10] Jan-Mark Geusebroek, Gertjan J. Burghouts, and Arnold W. M. Smeulders, "The amsterdam library of object images," *IJCV*, vol. 61, no. 1, pp. 103–112, 2005.